



# Architettura degli elaboratori - II

## CPU a ciclo singolo

Prof. Alberto Borghese  
Dipartimento di Scienze dell'Informazione  
[borgese@dsi.unimi.it](mailto:borgese@dsi.unimi.it)

Università degli Studi di Milano

Riferimento sul Patterson: capitolo 4.2 , 4.4, D1, D2.

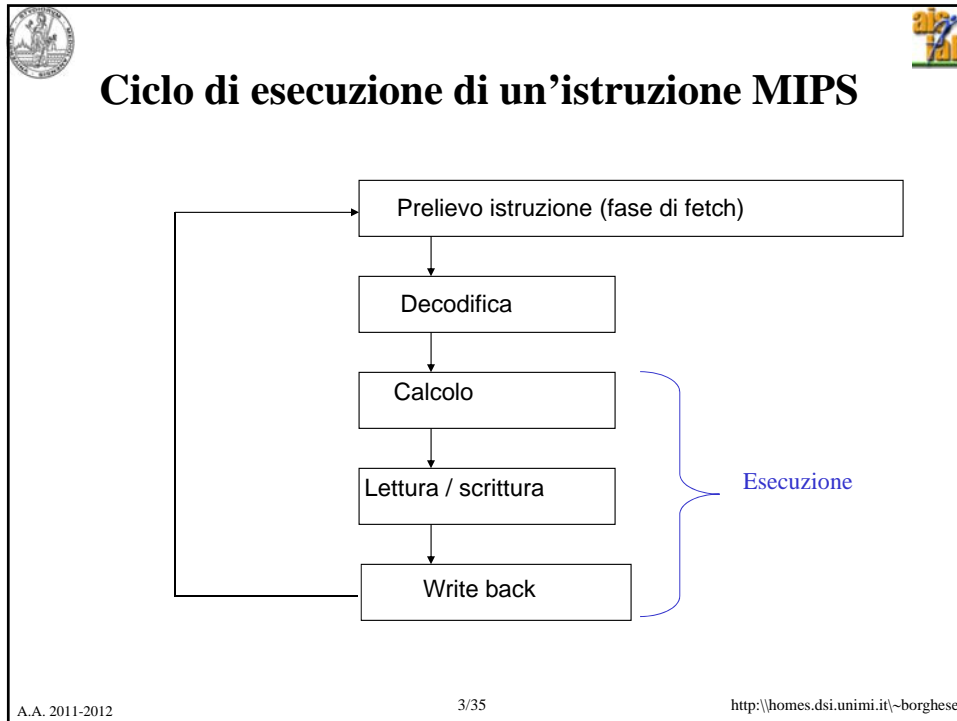


## Sommario

### La CPU

Costruzione di una CPU per le istruzioni di tipo R

Costruzione di una CPU per le istruzioni di tipo I (memoria).




**I componenti di un'architettura**

**CPU**


- Banco di registri (*Register File*) ad accesso rapido, in cui memorizzare i dati di utilizzo più frequente. Il tempo di accesso ai registri è circa 10 volte più veloce del tempo di accesso alla memoria principale.  
 Registro *Program counter (PC)*. Contiene l'indirizzo dell'istruzione corrente da aggiornare durante l'evoluzione del programma, in modo da prelevare dalla memoria la corretta sequenza di istruzione;  
 Registro *Instruction Register (IR)*. Contiene l'istruzione in corso di esecuzione. Questo registro verrà utilizzato più avanti nelle architetture multi-ciclo.
- Unità per l'esecuzione delle operazioni aritmetico-logiche (*Arithmetic Logic Unit - ALU*). I dati forniti all'*ALU* possono provenire da registri oppure direttamente dalla memoria, a seconda delle modalità di indirizzamento previste;
- Unità aggiuntive per elaborazioni particolari come unità aritmetiche per dati in virgola mobile (*Floating Point Unit - FPU*), sommatore ausiliari, ecc.;
- **Unità di controllo**. Controlla il flusso e determina le operazioni di ciascun blocco.

**MEMORIA PRINCIPALE**

A.A. 2011-2012 4/35 http://homes.dsi.unimi.it/~borgnese




## Codifica delle istruzioni




- Tutte le istruzioni MIPS hanno la **stessa dimensione (32 bit)** – Architettura RISC.
- I 32 bit hanno un significato diverso a seconda del formato (o tipo) di istruzione
  - il tipo di istruzione è riconosciuto in base al valore di alcuni bit (**6 bit**) più significativi (**codice operativo - OPCODE**)
- Le istruzioni MIPS sono di **3 tipi** (formati):
  - **Tipo R (register)** – Lavorano su **3 registri**.
    - Istruzioni aritmetico-logiche.
  - **Tipo I (immediate)** – Lavorano su **2 registri**. L'istruzione è suddivisa in un **gruppo di 16 bit contenenti informazioni + 16 bit riservati ad una costante**.
    - Istruzioni di accesso alla memoria o operazioni contenenti delle costanti.
  - **Tipo J (jump)** – Lavora **senza registri**: **codice operativo + indirizzo di salto**.
    - Istruzioni di salto incondizionato.

	6-bit	5-bit	5-bit	5-bit	5-bit	6-bit
R	op	rs	rt	rd	shamt	funct
I	op	rs	rt	indirizzo		
J	op	indirizzo				

A.A. 2011-2012
5/35
<http://homes.dsi.unimi.it/~borgnese>





## Istruzioni



<code>add \$s1, \$s2, \$s3</code>	00000	10010	10011	10001	00000	100000
<code>beq \$s1, \$s2, -100</code>	000100	10001	10010	1111	1111	1110 0111
<code>lw \$t0, 32 (\$s3)</code>	100011	10011	01000	0000	0000	0010 0000
<code>sw \$t0, 32 (\$s3)</code>	101011	10011	01000	0000	0000	0010 0000
<code>addi \$t0, \$s3, 64</code>	001000	10011	01000	0000	0000	0100 0000
<code>j 0x80000</code>	000010	00	0000	0100	0000	0000 0000

A.A. 2011-2012
6/35
<http://homes.dsi.unimi.it/~borgnese>





## Sommario

La CPU

Costruzione di una CPU per le istruzioni di tipo R

Costruzione di una CPU per le istruzioni di tipo I (memoria).

A.A. 2011-2012 7/35 http://homes.dsi.unimi.it/~borghese

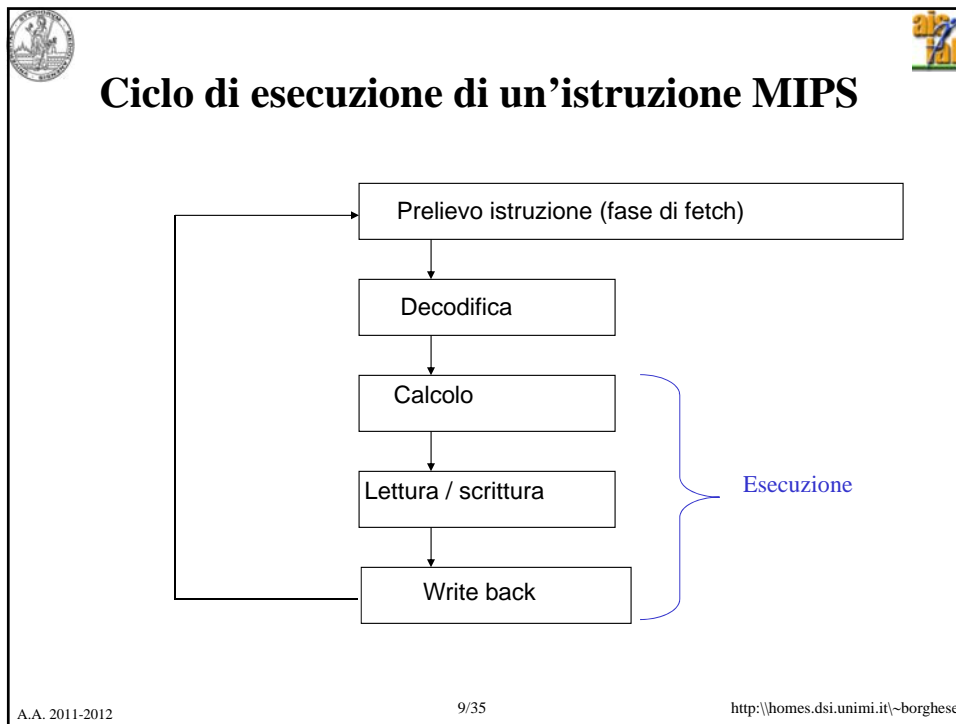


## Obiettivo

Costruzione di una CPU completa che sia in grado di eseguire:

- Accesso alla memoria in lettura (lw) o scrittura (sw).
- Istruzioni logico-matematiche (e.g. add, sub, and....).
- Istruzioni di salto condizionato (branch) o incondizionato (jump).

A.A. 2011-2012 8/35 http://homes.dsi.unimi.it/~borghese




## Come funziona una CPU?

- Usa un registro, il Program Counter (PC) per ottenere l'indirizzo dell'istruzione.
- Preleva l'istruzione dalla memoria e la inserisce nell'IR.
- Capisce di che tipo di istruzione si tratta (decodifica).
  - usa l'istruzione stessa per decidere cosa fare esattamente.
- Legge il contenuto dei registri.


Da qui le istruzioni si differenziano.

- Calcolo: utilizzo dell'ALU dopo aver letto i registri:
  - per calcolare l'indirizzo in memoria.
  - per eseguire un'operazione logico-aritmetica.
  - per effettuare test (uguaglianza, disuguaglianza, <...).
- Accesso alla memoria.
- Scrittura del risultato nel register file.

A.A. 2011-2012 10/35 http://homes.dsi.unimi.it/~borghese



## Come funziona una CPU?




- Usa un registro, il Program Counter (PC) per ottenere l'indirizzo dell'istruzione.
- Preleva l'istruzione dalla memoria e la inserisce nell'IR.
  
- Capisce di che tipo di istruzione si tratta (decodifica).
  - usa l'istruzione stessa per decidere cosa fare esattamente.
- Legge il contenuto dei registri.
  
- Da qui le istruzioni si differenziano.*
  
- Calcolo: utilizzo dell'ALU dopo aver letto i registri:
  - per calcolare l'indirizzo in memoria.
  - per eseguire un'operazione logico-aritmetica.
  - per effettuare test (uguaglianza, disuguaglianza, <...).
  
- Accesso alla memoria.
  
- Scrittura del risultato nel register file.


A.A. 2011-2012

11/35

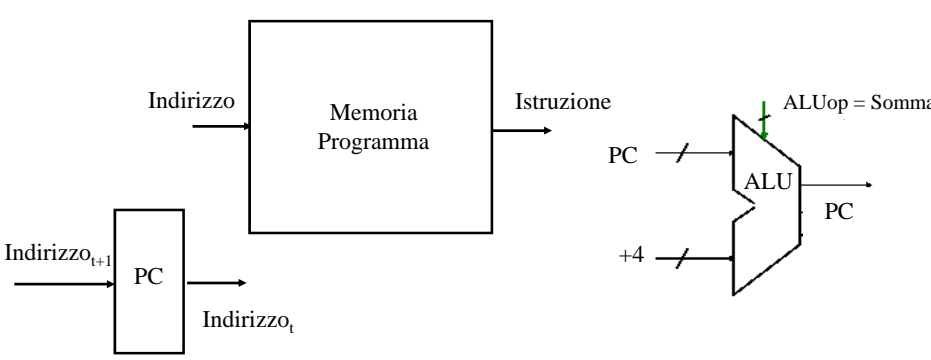
<http://homes.dsi.unimi.it/~borghese>



## Fase di fetch



- 1) Memorizzare l'indirizzo dell'istruzione nel PC.
- 2) Leggere l'istruzione dalla memoria.
- 3) Aggiornare l'indirizzo in modo che in PC sia contenuto l'indirizzo dell'istruzione successiva.



A.A. 2011-2012

12/35

<http://homes.dsi.unimi.it/~borghese>

## Temporizzazione

1 istruzione per ciclo di clock. Temporizzazione del PC.

$T > \text{Tempo necessario per eseguire il cammino critico}$

A.A. 2011-2012 13/35 http://homes.dsi.unimi.it/~borghese

## Circuito della fase di fetch



Clock

(Read)

ALUop = Somma

8000	sub \$s1, \$s2, \$s3
8004	add \$s4, \$s1, \$t1
....	

A.A. 2011-2012 14/3 http://homes.dsi.unimi.it/~borghese

## Istruzioni di tipo R

**R**



op = 0	rs = 18	rt = 19	rd = 17	Shamt=0	funct=34
6 bit	5 bit	5 bit	5 bit	5 bit	6 bit

sub \$s1, \$s2, \$s3

A.A. 2011-2012

15/35

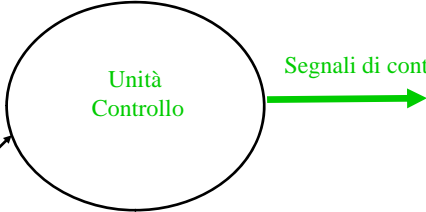
<http://homes.dsi.unimi.it/~borghese>

## Fase di decodifica

- 1) Leggo l'istruzione e genero i segnali di controllo opportuni.
- 2) Leggo il contenuto dei registri.

funct	6 bit
shamt	5 bit
rd	5 bit
rt	5 bit
rs	5 bit
op	6 bit



Unità  
Controllo

Segnali di controllo

A.A. 2011-2012

16/35

<http://homes.dsi.unimi.it/~borghese>



## Register file

Insieme di 32 registri da 32 bit

Contenuto 1  
Contenuto 2

#Reg read 1  
#Reg read 2  
#Reg write  
Contenuto Write

R W

Banco di registri utilizzabile come memoria  
Può essere scritto o letto.

Un mux per ogni porta di lettura.

Numero del registro letto 1 <sup>5</sup>/<sub>7</sub>

Numero del registro letto 2 <sup>5</sup>/<sub>7</sub>

Registro 0  
Registro 1  
...  
Registro n-1  
Registro n

MUX

MUX

Dato letto 1  
Dato letto 2

A.A. 2011-2012

17/35

<http://homes.dsi.unimi.it/~borgnese>

## Letture dei registri (istruzioni di tipo R)

- 1) Leggo l'istruzione e genero i segnali di controllo opportuni.
- 2) Leggo il contenuto dei registri.

func	6 bit
shamt	5 bit
rd	5 bit
rt	5 bit
rs	5 bit
op	6 bit

8000 sub \$s1, \$s2, \$s3  
8004 add \$s4, \$s1, \$t1  
.....

Insieme di 32 registri da 32 bit

Contenuto 1  
Contenuto 2

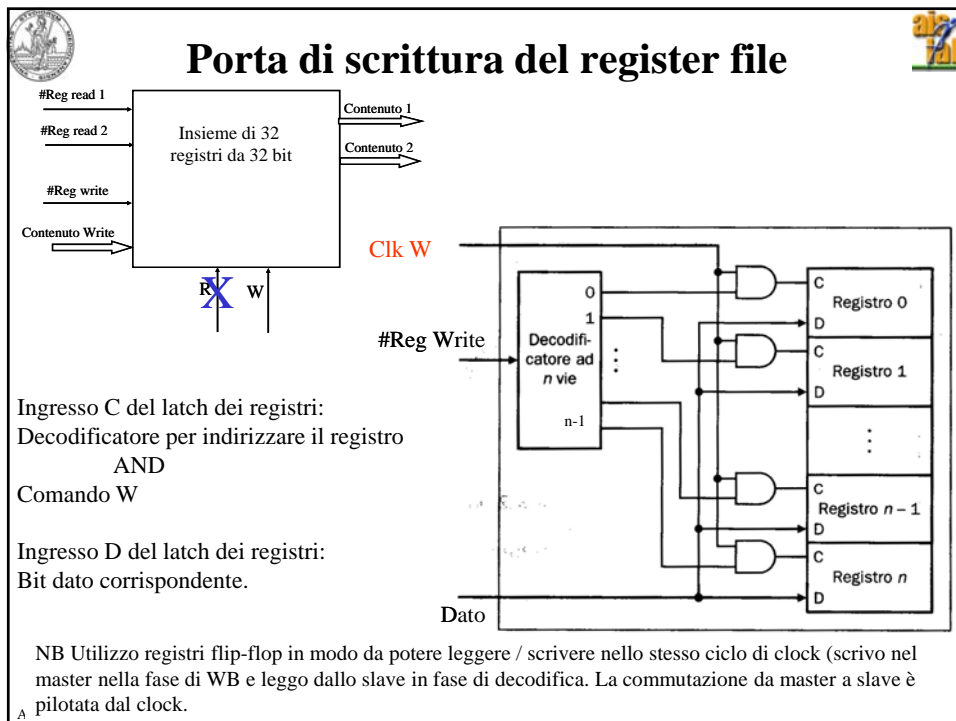
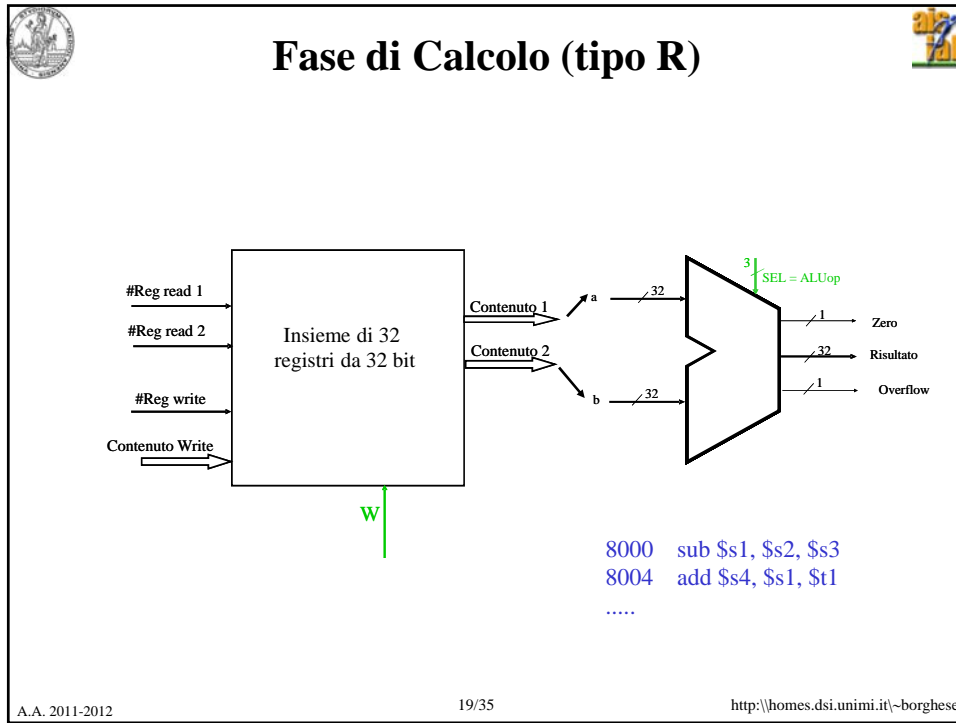
#Reg read 1  
#Reg read 2  
#Reg write  
Contenuto Write

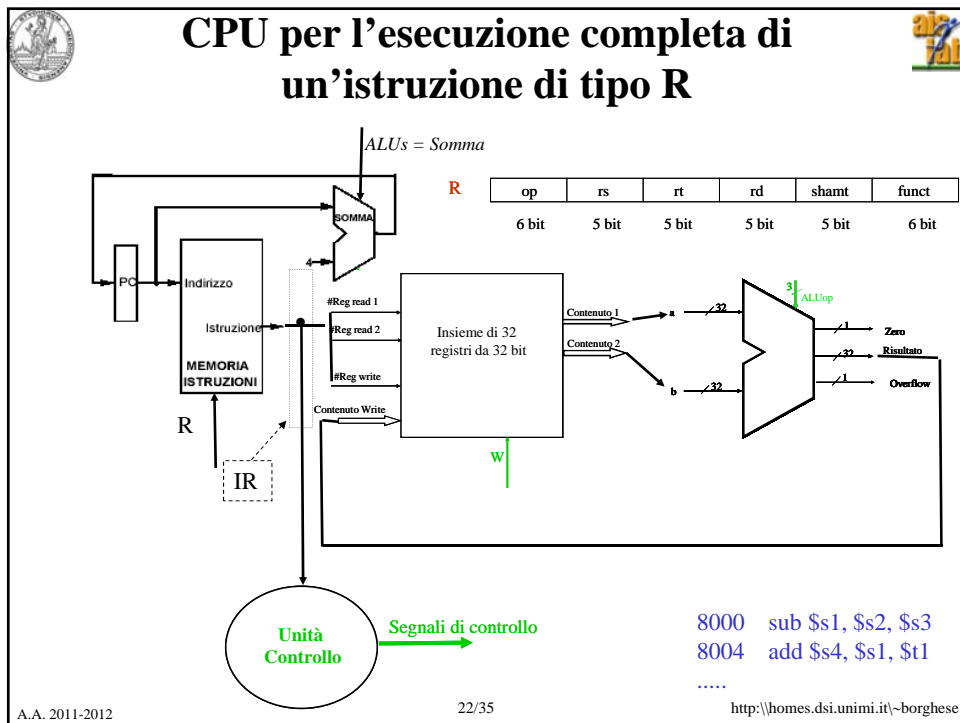
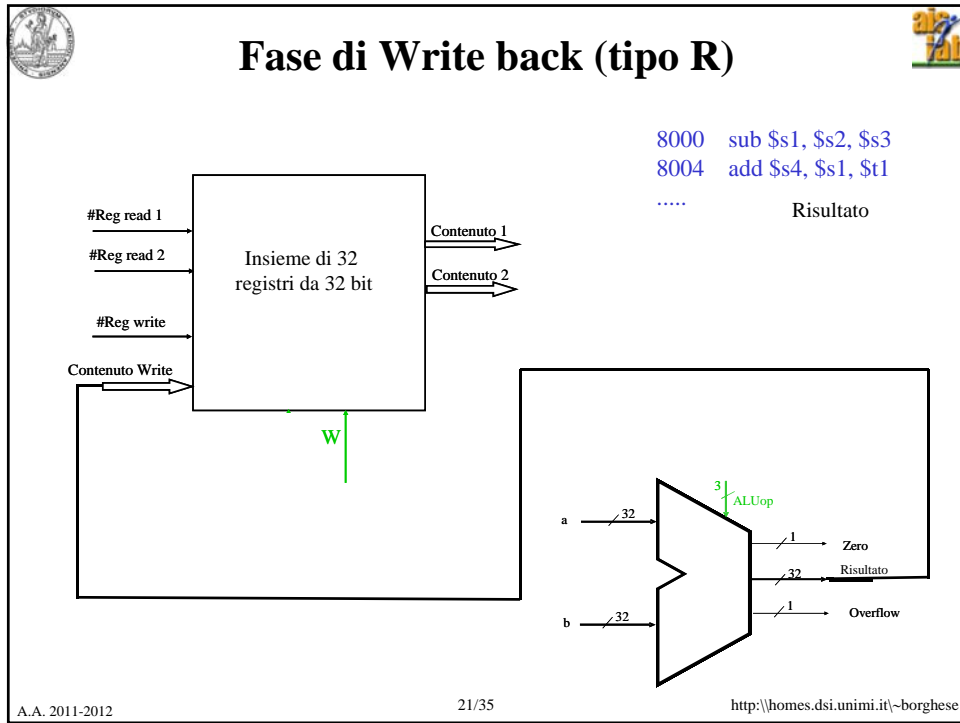
W



A.A. 2011-

18/35

<http://homes.dsi.unimi.it/~borgnese>





## Sommaro

La CPU



Costruzione di una CPU per le istruzioni di tipo R

Costruzione di una CPU per le istruzioni di tipo I (memoria).

A.A. 2011-2012

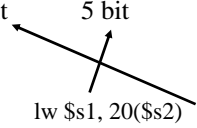
23/35

<http://homes.dsi.unimi.it/~borgnese>

## Istruzioni di tipo I: lw/sw

<b>I</b>	100011	10010	10001	0000 0000 0001 0100
	6 bit	5 bit	5 bit	16 bit



lw \$s1, 20(\$s2)

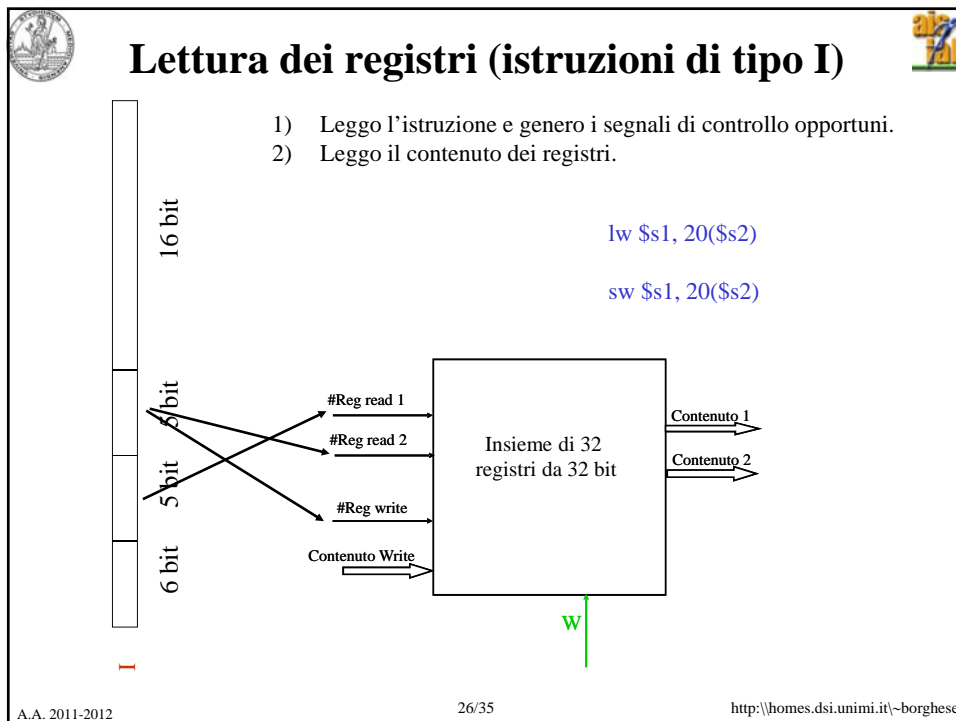
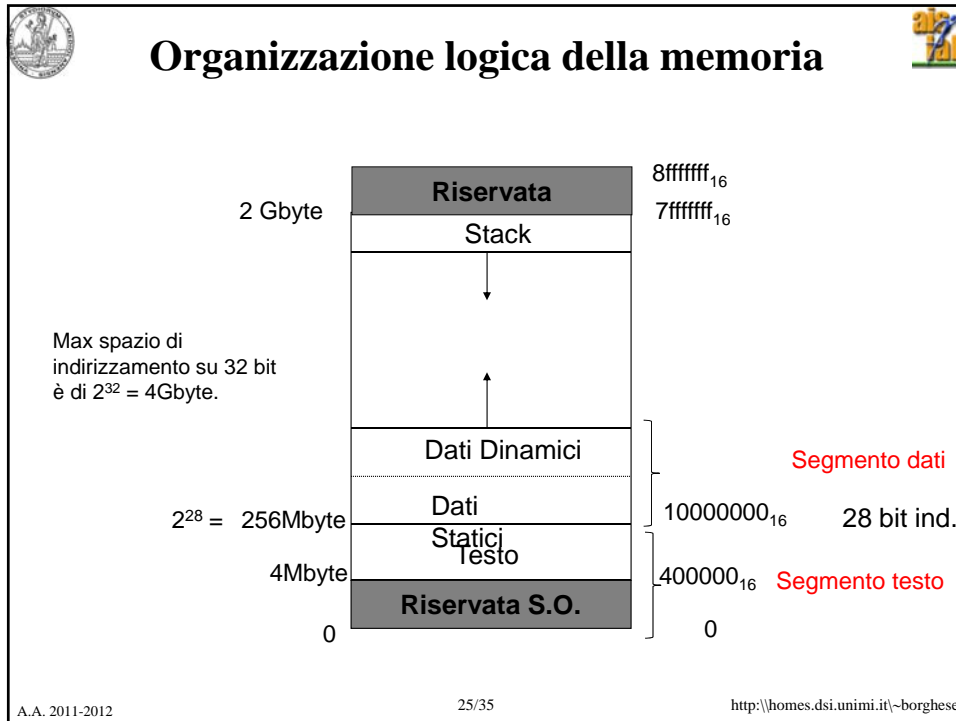
L'indirizzo di memoria sar :

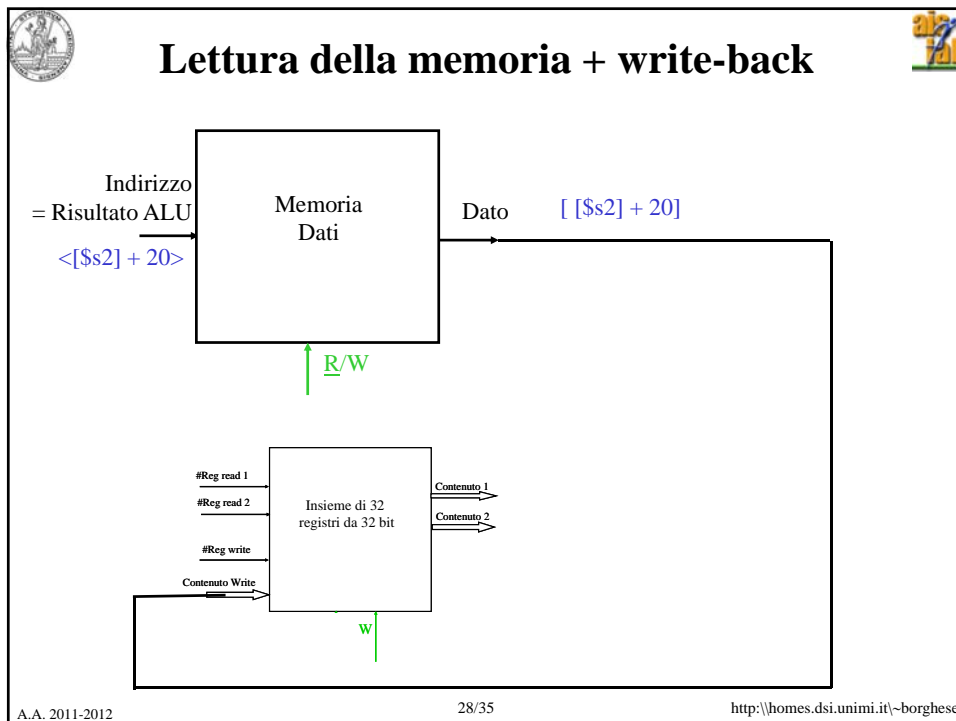
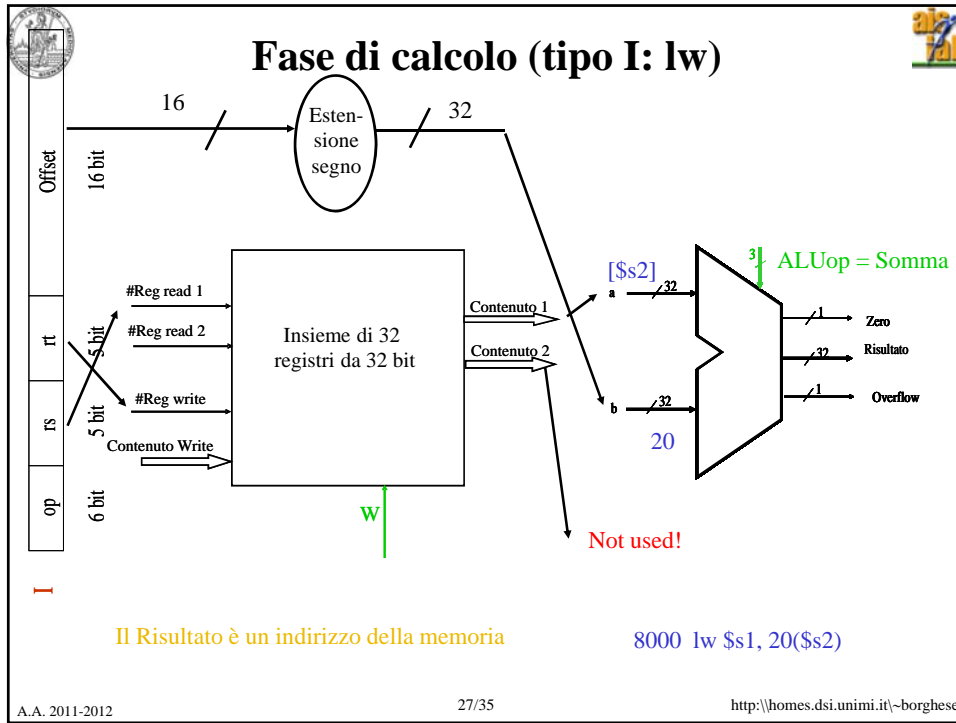
Base [\$s2]	0100 1000 0011 0001 1011 1011 1011 1011 +
Offset	0000 0000 0001 0100 +
<b>Indirizzo dato</b>	<b>0100 1000 0011 0001 1011 1011 1100 1111</b>

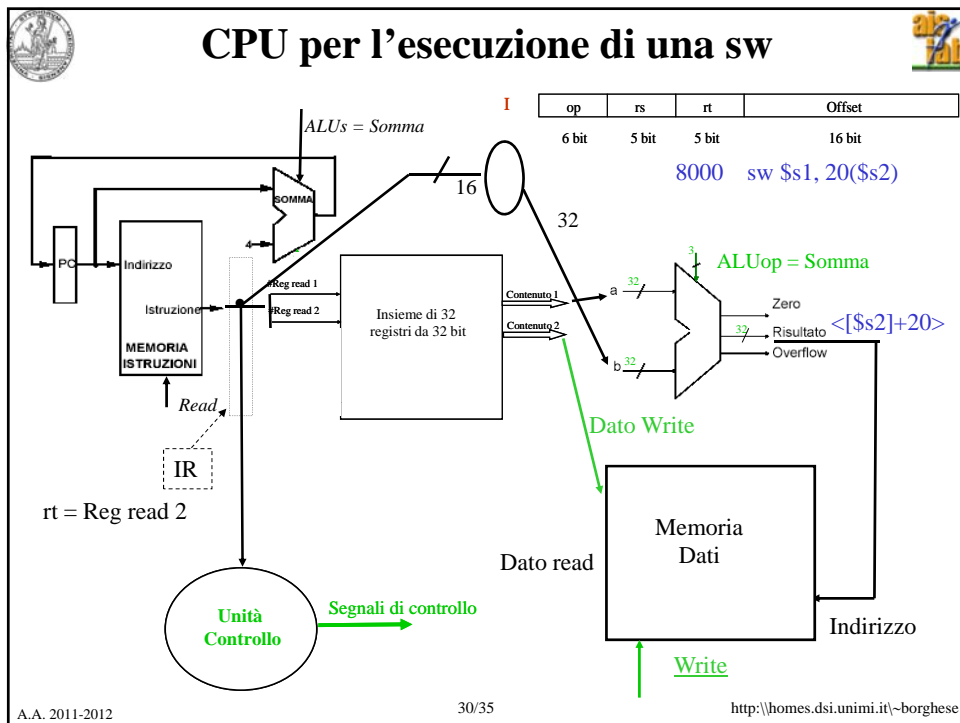
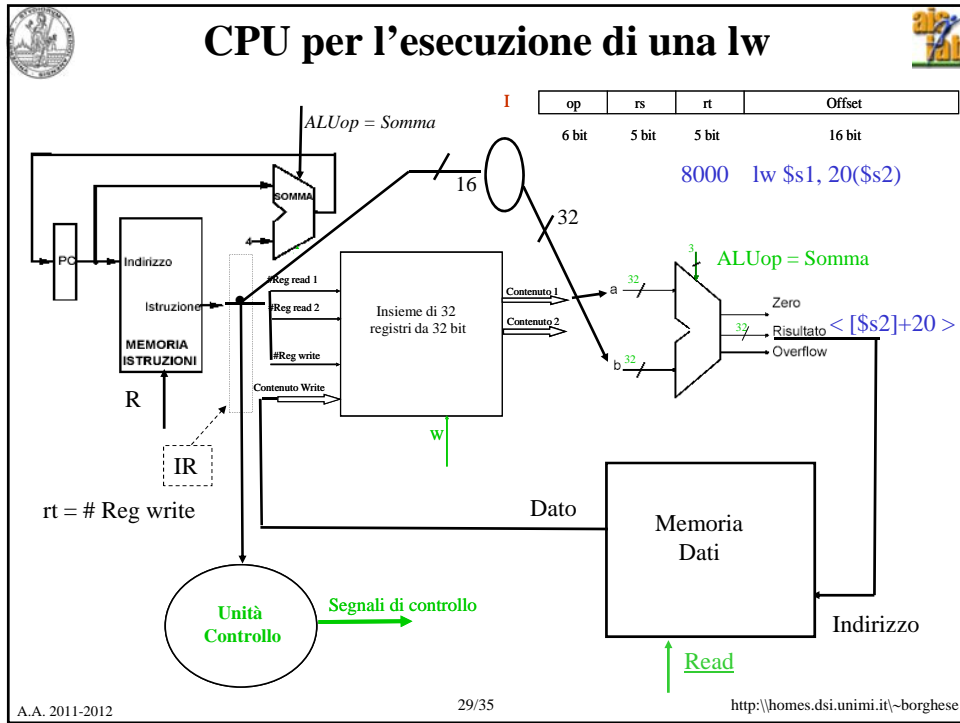
A.A. 2011-2012

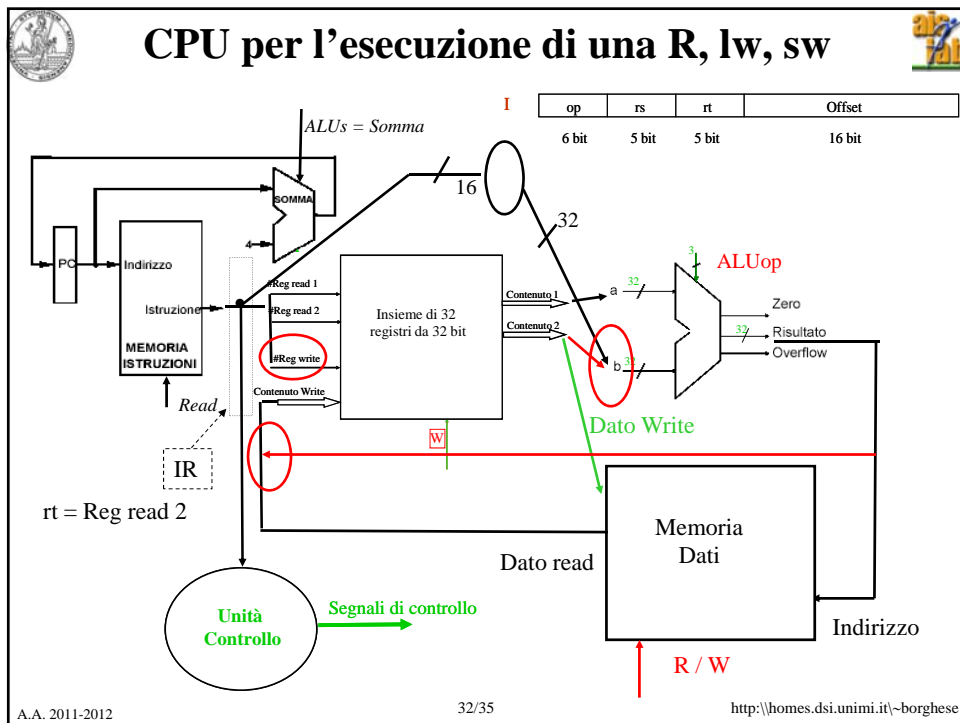
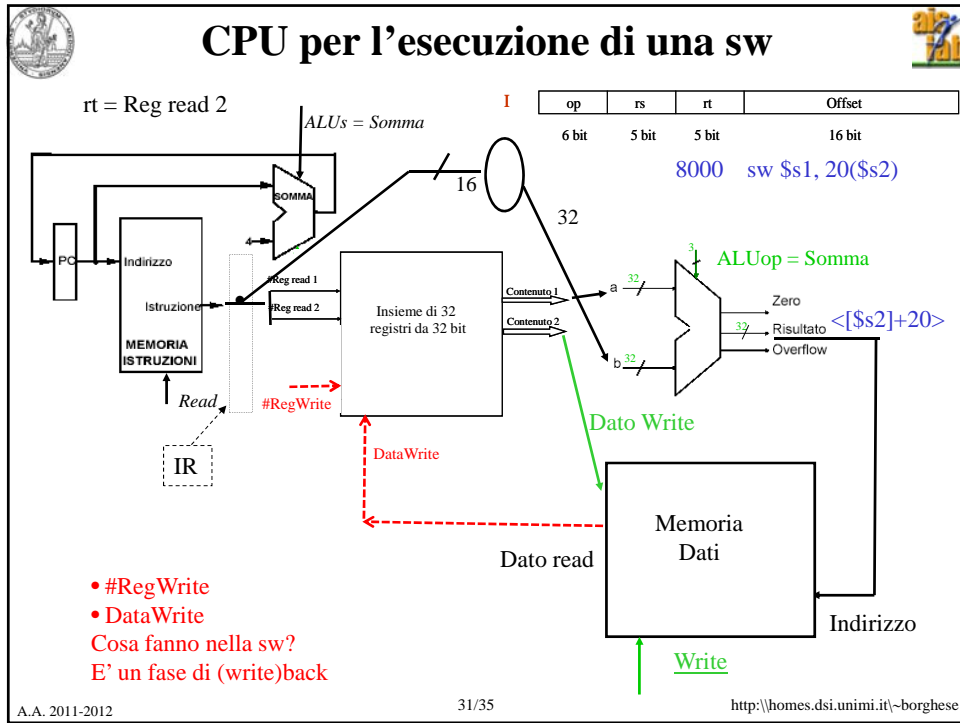
24/35

<http://homes.dsi.unimi.it/~borgnese>











## Rapporto UC - Dati

La CPU è un'architettura del tipo: Controllore - Data-path

*Fase comune nel ciclo di esecuzione:*

- Fase di fetch
- Decodifica (generazione dei segnali di controllo)

*Fase diversa: Esecuzione (Calcolo, Accesso memoria, WriteBack)*

A.A. 2011-2012 33/35 http://homes.dsi.unimi.it/~borghese

## L'unità di controllo

- Unità di controllo coordina i flussi di informazione (è il "cervello" della CPU):
- 1) abilitando le vie di comunicazione opportune a seconda dell'istruzione in corso di esecuzione.
- 2) selezionando l'operazione opportuna delle ALU.

A.A. 2011-2012 34/35 http://homes.dsi.unimi.it/~borghese



## Sommario



Costruzione di una CPU per le istruzioni di tipo R

Costruzione di una CPU per le istruzioni di tipo I (memoria).

Costruzione di una CPU per le istruzioni di tipo I (salti).